In the claims:

Please amend claims 1, 9, 14, 19 as follows:

5   1. (currently amended)   A domain-crossing verifier comprising:

a design scanner, receiving a textual design file specifying functions to be performed by a

chip being designed, for locating domain-crossing signals generated by a first

clock ~~but~~ and sampled by a second clock, wherein the first clock and the second

clock are asynchronous;

10   wherein the first clock has a first clock period and the second clock has a second clock

period that differs from the first clock period;

a delay randomizer that randomly selects as a random delay either a first delay value or a

second delay value, the first and second delay values differing by the second

clock period;

15   wherein the delay randomizer multiplies a random binary number by a period of the

second clock to generate the random delay; and

a delay applicator, coupled to the delay randomizer, for applying the random delay to a

first flip-flop, the first flip-flop being clocked by the second clock ~~but~~ and

receiving one of the domain-crossing signals generated by the first clock as an

20   input;

a simulator that simulates the textual design file using the random delays applied by the

delay applicator to simulate logic hazards caused by domain-crossing signals;

a reporter for reporting machine-generated results of the simulator to a user on a display

or in a machine-generated report;

25   wherein the delay applicator applies a series of random delays generated by the delay

randomizer to a plurality of the domain-crossing signals located by the design

scanner,

wherein the chip defined by the textual design file ~~can be~~ is simulated using the random

delays that differ by the second clock period to simulate logic hazards caused by domain-

30   crossing signals.

2. (original)   The domain-crossing verifier of claim 1 wherein each domain-crossing

signal passes through a synchronizer that includes the first flip-flop clocked by the

second clock and generating a middle signal, and a second flip-flop that receives

the middle signal and is clocked by the second clock to generate a re-

5   synchronized signal that can be sampled by logic in a second domain clocked by

the second clock;

wherein the synchronizers are added to the textual design file by the design scanner or are

already part of the textual design file.

10   3. (original)   The domain-crossing verifier of claim 2 further comprising:

a cycle simulator, performing the functions defined by the textual design file on input

stimuli, the cycle simulator delaying sampling of the domain-crossing signals by

the second clock by the random delays generated by the delay randomizer,

wherein the chip defined by the textual design file is simulated by the cycle simulator

15   using the random delays that differ by the second clock period to simulate logic hazards

caused by domain-crossing signals.

4. (original)   The domain-crossing verifier of claim 3 wherein the delay applicator adds

delay parameters specifying the random delay generated by the delay randomizer to

20   statements in the textual design file for statements that define the first flip-flop of

synchronizers of domain-crossing signals.

5. (original)   The domain-crossing verifier of claim 4 further comprising:

a compiler, reading the textual design file, for checking statements in the textual design

25   file for proper syntax, and for generating a netlist defining the chip to be designed

for input to the cycle simulator during simulation.

6. (original)   The domain-crossing verifier of claim 1 wherein the first delay value is

zero and the second delay value is the second clock period, or wherein the first delay

30   value is an arbitrary delay value and the second delay value is the arbitrary delay value

added to the second clock period.

7. (canceled)

8. (original)   The domain-crossing verifier of claim 1 wherein the delay randomizer

5   receives a seed value that specifies a starting point in a random-number sequence for

generating the random delays.

9. (currently amended)   A method for simulating domain-crossing signals that cross

from a first clock domain to a second clock domain comprising:

10   identifying a domain-crossing signal generated by a first clock in the first clock domain,

        ~~but~~ and sampled by a second clock in the second clock domain;

inserting an added delay to sampling of the domain-crossing signal by the second clock,

        wherein the added delay is selected from a first delay and a second delay, wherein

        the second delay is a period of the second clock greater than the first delay;

15   wherein inserting an added delay comprises multiplying a binary random number with

        the period of the second clock to generate the added delay;

wherein the steps of identifying a domain-crossing signal and inserting the added delay

        are repeated for other domain-crossing signals, wherein some domain-crossing

        signals have the second delay selected as the added delay while other domain-

20   crossing signals have the first delay selected as the added delay;

simulating a design containing the domain-crossing signals having the added delays to

        sampling by the second clock; and

reporting machine-generated simulation results to a user on a display or on a machine-

        generated report,

25   whereby the design is simulated with added delays on domain-crossing signals wherein

the added delays differ by the period of the second clock.

10. (original) The method of claim 9 wherein simulating the design comprises

        simulating from a design-language file before logic gates are synthesized,

30   whereby domain-crossing signals are verified before gate-level synthesis.

11. (original) The method of claim 9 wherein simulating the design comprises

        simulating from a design-language file before physical layout of logic gates and

        wiring-connection routes are generated by a routing program,

whereby domain-crossing signals are verified before layout and routing.

5

12. (original) The method of claim 9 wherein the steps of identifying the domain-

crossing signal and inserting the added delay are repeated for all domain-crossing signals.

13. (original) The method of claim 12 wherein inserting an added delay comprises

10   randomly selecting the first delay or the second delay as added delay,

whereby added delays are selected randomly.

14. (currently amended)   The method of claim 12 wherein for each added delay for

        each domain-crossing signal, the first delay is simulated during one simulation

15   run, ~~but~~ and the second delay is selected as the added delay during another

        simulation run,

whereby added delays are exhaustively simulated for domain-crossing signals.

15. (canceled)

20

16. (original) The method of claim 12 further comprising:

inserting a synchronizer onto the domain-crossing signal, wherein the synchronizer

        comprises a first flip-flop and a second flip-flop in series;

wherein inserting an added delay comprises adding the added delay to an output of the

25   first flip-flop before an input to the second flip-flop,

whereby synchronizer delays are randomized.

17. (original) The method of claim 12 wherein the first clock is asynchronous to the

second clock.

30

18. (original) The method of claim 12 further comprising:

identifying a multi-cycle signal generated by the second clock in the second clock

domain, wherein the multi-cycle signal is allowed more than one period of the

second clock to propagate before being sampled by the second clock;

inserting an added delay to sampling of the multi-cycle signal by the second clock,

5   wherein the added delay is selected from the first delay and the second delay,

wherein the second delay is a period of the second clock greater than the first

delay;

wherein simulating the design includes simulating the multi-cycle signals having the

added delays to sampling by the second clock;

10   whereby the design is simulated with added delays on multi-cycle signals wherein the

added delays differ by the period of the second clock.

19.   (currently amended)   A domain-crossing signal verifier comprising:

identifying means for identifying a domain-crossing signal generated by a first clock in a

15   first clock domain, ~~but~~ and sampled by a second clock in a second clock domain;

wherein the domain-crossing signal is generated by the first clock passes through a first

flip-flop and a second flip-flop in a synchronizer, the first and second flip-flops

clocked by the second clock;

delay randomizer means for generating a randomized delay for the first flip-flop in the

20   synchronizer, wherein the randomized delay is randomly selected as either a first

delay or a second delay wherein the second delay is the first delay added to a

period of the second clock, the delay randomizer means also for multiplying a

binary random number with the period of the second clock to generate the

randomized delay;

25   repeat means for activating the delay randomizer means to generate randomized delays

for other domain-crossing signals identified by the identifying means;

wherein some domain-crossing signals have the second delay selected as the randomized

delay while other domain-crossing signals have the first delay selected as the

randomized delay;

simulating means for simulating a design containing the domain-crossing signals having

the randomized delays for first flip-flops in synchronizers on the domain-crossing

signals before sampling by the second clock; and

report means for reporting machine-generated results from the simulating means to a user

5   on a display or on a machine-generated report,

whereby the design is simulated with randomized delays on domain-crossing signals

wherein the randomized delays differ by the period of the second clock.

20. (original) The domain-crossing signal verifier of claim 19 further comprising:

10   multi-cycle means, receiving a list of multi-cycle signals that are allowed more than one

period of the second clock for signal propagation, for activating the delay

randomizer means to generate randomized delays for a gate in a path of each of

the multi-cycle signals on the list of multi-cycle signals;

wherein the randomized delays are applied to multi-cycle signals for simulation by the

15   simulating means.